

## 1. primary data type

1. integers/real number

2. float/double (the difference between these two is they have different numerical digits)

3. Char

4. boolean

## 2. abstract data structures

### 1. 2D array

```
int[][] twoD = new int[4][5];
```

### 2. Stack

Last in, first out

All about enqueue and dequeue

3 stack methods:

Method name	Brief description	Example: OPS, a stack of Integers	Comment
push()	Push an item onto the stack	OPS.push(42)	Adds an element that contains the argument, whether it is a value, String, object, etc. to the top of the stack.
pop()	Pop an item off the stack	NUM = OPS.pop()	Removes and returns the item on the top of the stack.
isEmpty()	Test: stack contains no elements	if OPS.isEmpty() then	Returns TRUE if the stack does not contain any elements.

### 3. Queue

First in, first out

3 queue methods:

Method name	Brief description	Example: WAIT, a queue of Strings	Comment
enqueue()	Put an item into the end of the queue	WAIT.enqueue("Mary")	Adds an element that contains the argument, whether it is a value, String, object, etc. to the end of the queue.
dequeue()	Remove an item from front of the queue	CLIENT = WAIT.dequeue()	Removes and returns the item at the front of the queue.
isEmpty()	Test: queue contains no elements	if WAIT.isEmpty() then	Returns TRUE if the queue does not contain any elements.

#### 4. Linked list

1. A linked list is a linear collection of self-referential structures, called nodes, connected by pointer links.

2. A linked list is accessed by keeping a pointer to the first node of the list.

3. This pointer to the first node of a list is typically named head.

4. Subsequent nodes are accessed via a link pointer member that is stored in each node.

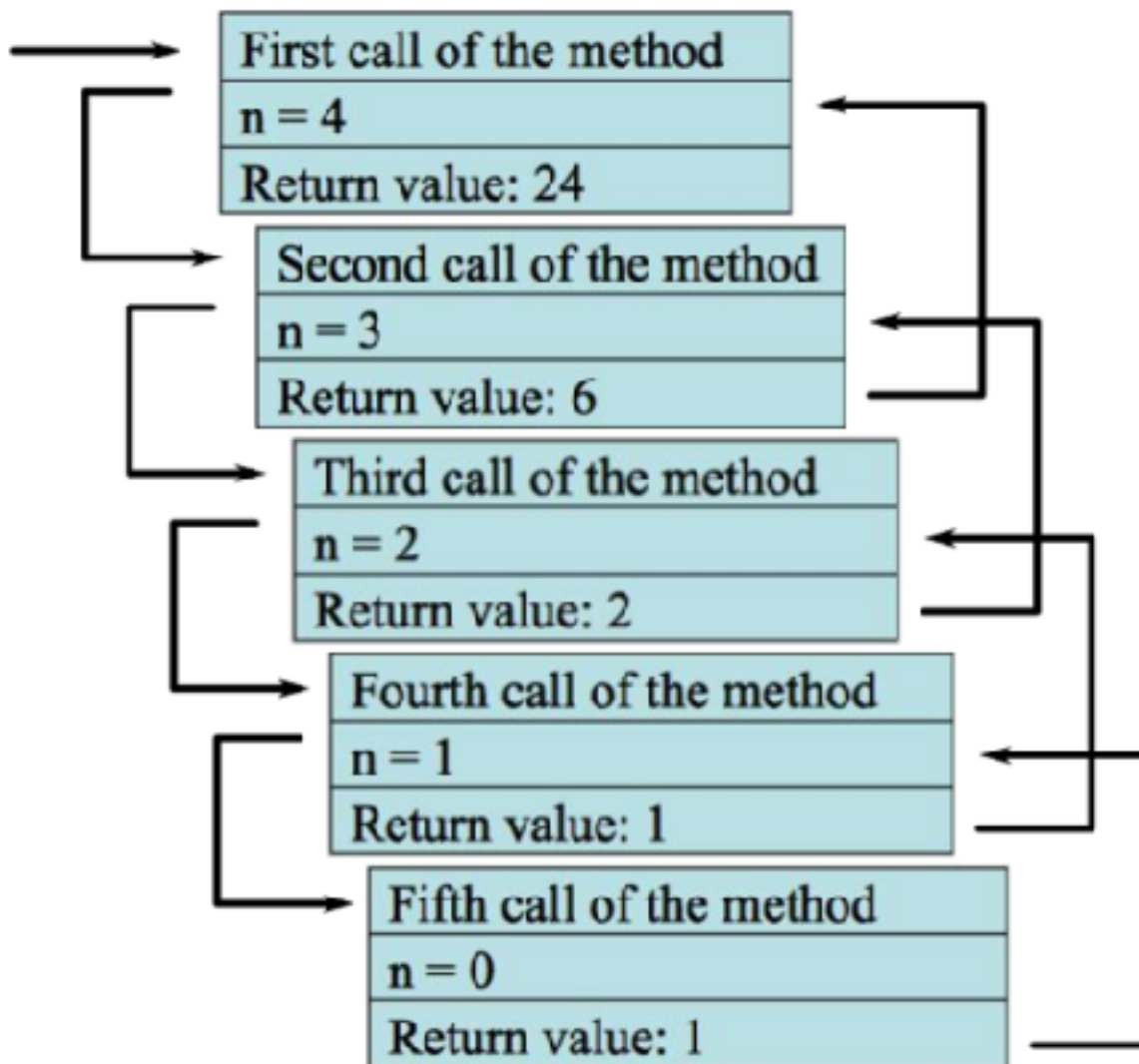
#### 5. Tree

a binary tree is either empty (represented by a null pointer), or is made of a single node, where the left and right pointers (recursive definition ahead) each point to a binary tree.

Recursion:

1. a method where the solution to a problem depends on solutions to smaller instance of the same problem.

2. a method that call itself (with a stopping condition)



Recursion Vs. Iteration:

recursion

1. low efficiency
2. easy to go out of bound

iteration

1. high efficiency
2. complicate code

Arrays in general :

```
int[] num={3,11,9,74,8,2,18,71,43,10};  
int[] nu=new int[]{3,11,9};
```